Reconhecimento Automático de Caracteres Braille Utilizando Rede Neural Convolucional YOLO

Roberto P. Nascimento, Universidade Federal do Rio Grande do Sul (UFRGS) / PPGIE
robertotpd@gmail.com, https://orcid.org/0000-0002-2909-3189
Josivan R. Reis, Universidade Federal do Oeste do Pará (UFOPA),
josivanrreis@gmail.com, https://orcid.org/0000-0002-0949-4858
Lorenzo A. F. Aragão, Universidade Federal do Oeste do Pará (UFOPA),
lorenzoalberto02@gmail.com, https://orcid.org/0009-0004-0705-6442
Arthur S. Araújo, Universidade Federal do Rio Grande do Sul (UFRGS) / PPGIE,
arthuraraujoaraujo1@gmail.com, https://orcid.org/0000-0002-9905-6850
Regina Heidrich, Universidade Feevale,
rheidrich@feevale.br, https://orcid.org/0000-0001-9101-1124
Dante A. C. Barone, Universidade Federal do Rio Grande do Sul (UFRGS),
barone@inf.ufrgs.br, http://orcid.org/0000-0002-5133-0144

Resumo: O Braille é um sistema essencial na formação dos alunos com deficiência visual, mas o desconhecimento desse sistema por parte do professor tem gerado dificuldade na comunicação escrita com os alunos. Assim, para ajudar a superar esse desafio, esta pesquisa apresenta uma nova abordagem para o reconhecimento automático de caracteres Braille utilizando rede neural convolucional YOLO. A novidade deste trabalho está na tradução de caracteres Braille para Português e na investigação sobre a influência de ruídos e rotações na detecção do Braille. O método projetado é robusto e capaz de identificar os caracteres Braille com precisão acima de 97%

Palavras-chave: Braille, Reconhecimento de Caracteres, YOLO.

Automatic Braille Character Recognition using YOLO Convolutional Neural Network

Abstract: Braille is an essential system in the education of visually impaired students, but teachers' lack of knowledge of this system has created difficulties in written communication with students. Therefore, to help overcome this challenge, this research presents a new approach to automatic Braille character recognition using the YOLO Convolutional Neural Network. The novelty of this work is the translation of Braille characters into Portuguese and the investigation of the influence of noise and rotations on Braille recognition. The developed method is robust and able to identify Braille characters with an accuracy of more than 97%.

Keywords: Braille, Character Recognition, YOLO.

1. Introdução

A comunicação inclusiva cria condições para que a sociedade desperte para o fato de que, se tratando de pessoas, o que importa é sua capacidade de realização, evolução, produção, e não suas limitações. A educação inclusiva deve garantir a participação de qualquer aluno no processo de ensino e de aprendizagem nas salas de aula regular (Leite, 2021). Mas para que a inclusão no âmbito educacional ocorra de forma satisfatória, os professores videntes*das escolas de ensino regular, deverão estar aptos para atender a comunicação escrita dos alunos com deficiência visual (DV).

Em (Hsu, 2020) é mencionado o desequilíbrio entre os métodos de comunicação que possa permitir videntes a entender Braille e, para isso, é necessário desenvolver

*Termo usado para descrever pessoas que não têm limitações visuais graves.

tecnologias que ajudem na comunicação escrita entre estudantes com DV e professores videntes, pois conforme a Lei Brasileira de Inclusão da Pessoa com Deficiência[†] no Brasil, todas as crianças que estão em idade escolar, sem exceção, devem frequentar as salas de aulas do ensino regular.

Com isso, há uma lacuna na comunicação escrita entre professores videntes e alunos com DV, também entre os seus pares, gerando dificuldades no processo de ensino e de aprendizagem. Assim, este trabalho tem como objetivo construir um método capaz de reconhecer Braille, a partir de imagens, apresentando o caractere correspondente em português. O estudo busca desenvolver um modelo de reconhecimento automático que possa auxiliar os professores videntes, envolvidos na educação de crianças com DV, na tradução Braille. O restante deste trabalho está organizado da seguinte forma: A Seção 2 apresenta os trabalhos correlatos. Os materiais e métodos utilizados são mostrados na Seção 3. A análise dos resultados são descritos na Seção 4. Por fim, a Seção 5, apresenta as considerações finais.

2. Trabalhos Correlatos

Com foco em superar as dificuldades de comunicação escrita entre deficientes visuais e videntes, são encontradas na literatura alguns trabalhos baseados em OBR (*Optical Braille Recognition*) e CNN (rede neural convolucional) para converter o texto Braille para texto literal, realizando o reconhecimento automático do Braille (Ovodov, 2021; Li e Yan, 2021; Prakash; Thomas e Gopalan, 2020; Vasanthapriyan e Silva, 2019; Alufaisan *et al.*, 2021).

Em (AlSalman *et al.*, 2021) é proposto uma abordagem baseada em *deep learning* para converter imagens de Braille em textos multilíngues (Árabe, Inglês, Bangladesh e Índia), utilizando sistema OBR. Os autores desenvolveram um modelo de rede neural convolucional profunda (*deep convolutional neural network* - DCNN) e avaliaram em dois *datasets* de tamanhos e quantidades de caracteres distintos. O modelo proposto obteve uma precisão de classificação de 99,28% e 98,99% no conjunto de teste de cada *dataset*. No entanto, a abordagem não é estendida para cobrir abreviações/contrações em Braille na etapa de mapeamento multilíngue e não detecta automaticamente o idioma do documento em Braille.

No trabalho (Hsu, 2020), é empregada um modelo CNN e algoritmo de segmentação RCSA (*ratio character segmentation algorithm*) para reconhecimento e tradução do caractere Braille para o Inglês, tendo o desempenho do modelo CNN uma precisão de previsão de 98,73% no conjunto de teste. Contudo, a abordagem apresenta a limitação de converter linha por linha do texto Braille.

Uma abordagem em etapas utilizando CNN para o reconhecimento de caracteres Braille para inglês e chinês foi proposta em (Kausar *et al.*, 2021). Os autores fizeram alguns ajustes nos módulos da CNN para tornar a rede mais leve e melhorar desempenho de reconhecimento, através da estratégia de substituir alguns módulos das CNNs originais por um módulo de menor custo computacional, IRB (*inverted residual block*). Com isso, obteve uma precisão de previsão de 95,2% e 98,3%, nos *datasets* Braille em inglês e chinês, respectivamente.

Até onde foi possível pesquisar, este trabalho é o primeiro a realizar o reconhecimento de caracteres em Braille com a correspondente em português utilizando a rede neural convolucional YOLO (*You Only Look Once*). Através da análise experimental verificou-se a precisão de reconhecimento, tendo como principais contribuições: (i) Uma proposta para reconhecimento automática de caracteres Braille apresentando a sua

[†]https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2015/lei/l13146.htm

correspondente em Português; (ii) Introdução de uma investigação sobre a influência de ruídos e rotações na detecção e tradução de caracteres Braille. Essa investigação propicia uma compreensão sobre a análise das rotações nas imagens e, em quais casos sua aplicação excessiva pode comprometer o desempenho da leitura em relação ao modelo escolhido; (iii) O modelo proposto utiliza o algoritmo YOLO versão 5 (YOLOv5) pelo seu diferencial na combinação entre desempenho e acurácia, proporcionando detecção rápida e eficiente dos caracteres Braille.

Assim como (Kausar *et al.*, 2021), verificou-se que há um número limitado de *datasets* públicos disponíveis, o que dificulta uma comparação exata entre os diferentes algoritmos utilizados nos trabalhos correlatos. Dessa forma, como a grande maiorias dos trabalhos mencionados usam *datasets* privados para o reconhecimento das imagens em Braille, não é trivial realizar uma comparação justa entre esses resultados.

Na Tabela 1, é apresentada uma comparação entre as principais características dos trabalhos correlatos, com base nos seguintes critérios: objetivos, dados utilizados, disponibilidade do *dataset*, técnicas utilizadas e idioma.

3. Materiais e Métodos

O diagrama com a visão geral da abordagem proposta é apresentado na Figura 1, com as seguintes etapas: imagens Braille, pré-processamento das imagens, modelo YOLOv5, ajuste dos hiperparâmetros e reconhecimento dos caracteres Braille.

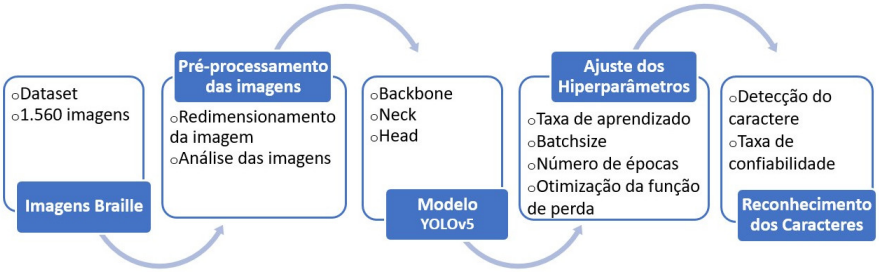


Figura 1. Visão geral do processo de reconhecimento de caracteres Braille

A etapa 1 é usada para aquisição dos caracteres Braille do *dataset* público da plataforma Kaggle (Shanks0465, 2020). Até onde foi possível pesquisar, este *dataset* é o único disponível publicamente com caracteres Braille rotulado, de grau 1, ou seja, que não é contraído, o que significa que há um caractere Braille específico para cada letra, totalizando 1.560 imagens das 26 letras do alfabeto, de ”a” a ”z”, com imagem de 28x28 pixels em escala de cinza.

Na etapa de pré-processamento, as imagens de entrada foram redimensionadas para 320x320 *pixels* e foi criado dois novos *datasets*, a partir da análise das imagens do original, aplicando um ajuste na rotação das imagens de até 45 graus, para um dos *datasets*.

Na etapa 3, a arquitetura do modelo selecionado foi YOLOv5, variação Small que apresenta 3 subdivisões principais da rede: o (i) *Backbone*, responsável por extrair características (*features*) da imagem por convolução múltipla e agrupamento, (ii) a *Neck*, a qual trata essas *features* para a etapa de detecção, agregando e combinando as mesmas e (iii) o *Head*, que é responsável pela parte final da detecção, possuindo como saída um vetor contendo os dados das detecções feitas (descrições das caixas delimitadoras) e, com isso, a imagem com a localização dos objetos de interesse é formado, como mostra a Figura 2.

Na etapa 4, ocorre o ajuste dos principais hiperparâmetros utilizados na arquitetura

Tabela 1. Tabela comparativa dos trabalhos correlatos

Referência	**Objetivo**	**Dados Utilizados**	**Disponibilidade do Dataset**	**Técnicas Utilizadas**	**Idioma**
(Alufaisan *et al.*, 2021)	Desenvolver um sistema de detecção e reconhecimento de sistema de numeração arábico baseado em *Deep Learning*.	Imagens para treinamento (3.000), validação (1.000) e teste (1.000).	Não disponível	ResNet	Árabe
(Li e Yan, 2021)	Implementar um tradutor Braille baseado em caracteres usando ResNet (*Residual Network*).	Imagens para treinamento (2.248), validação (280) e teste (280).	Não disponível	ResNet(ResNet-18, ResNet-34 e ResNet-50) e ABCNet (*Adaptive Bezier-Curve Network*)	Inglês
(Ovodov, 2021)	Fornecer um método de reconhecimento de imagens de texto em Braille obtidas por uma câmera de celular, por meio de OBR e CNN.	Imagens para treinamento (191) e teste (49).	Disponível	RetinaNet	Russo
(Prakash; Thomas e Gopalan, 2020)	Implementar a conversão de Braille para o Inglês utilizando *deep learning*.	Imagens para treinamento (1.560). Imagens para teste não informado.	Não disponível	Pré-processamento, segmentação e CNN	Inglês
(Vasanthapriyan e Silva, 2019)	Desenvolver uma plataforma para converter Braille para Cingalês para formalizar a comunicação escrita entre DV e pessoas videntes.	Um scanner foi utilizado para aquisição de 12 folhas de manuscrito Braille, com resolução de 200 dpi e 300 dpi.	Não disponível	Técnicas de processamento de imagem	Cingalês
(AlSalman *et al.*, 2021)	Propor uma abordagem baseada em *deep learning* para converter imagens com texto Braille em textos multilíngues.	Utiliza 2 *datasets*, sendo imagens para treinamento (1.124 e 3.165), validação (144 e 395) e teste (144 e 1.860).	Não disponível	OBR e CNN	Multilíngue
(Hsu, 2020)	Desenvolver uma abordagem para converter imagens Braille em texto em Inglês.	*Dataset* com 26.724 imagens, sendo 80% para treinamento e 20% dividido igualmente para validação e teste.	Não disponível	Pré-processamento, segmentação (RCSA) e CNN	Inglês
(Kausar *et al.*, 2021)	Desenvolver uma abordagem de reconhecimento automático de caracteres em Braille com foco no custo computacional.	Utilizou 2 *datasets*, com 1.560 e 144 imagens cada, sendo imagens para treinamento (1.092 e 70), validação (156 e 14) e teste (312 e 30), respectivamente.	Disponível	Pré-processamento e CNN (VGG16, ResNet50, InceptionV3 e DensNet201)	Inglês e Chinês
Este Trabalho	Propor um modelo de reconhecimento automático do Braille com a letra correspondente em português	Imagens para treinamento (1.092), validação (312) e teste (156).	Disponível	YOLO	Português

convolucional do YOLOv5, sendo definidas a taxa de aprendizado de 0.01, o momento igual a 0.937 e o decaimento de peso (*weight decay*) de 0.0005. Em seguida, definiu-se o *batchsize* (tamanho de lote) igual a 16, com número de épocas igual a 200. Para a otimização da função de perda, foi escolhido o otimizador SGD (*Stochastict Gradient Descent*). Por fim, na etapa 5, é apresentada as detecções dos caracteres Braille.

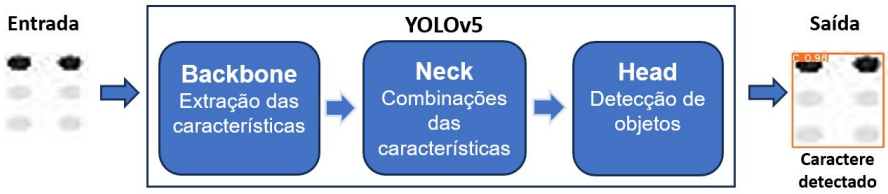


Figura 2. Arquitetura do modelo escolhido. Adaptado de (Kubera *et al.*, 2022; Albuquerque *et al.*, 2022)

3.1. Configurações

A etapa de treinamento foi realizada na plataforma online Google Colab[‡], a qual disponibiliza uma máquina virtual com placa de vídeo dedicada (GPU), NVIDIA Tesla T4 12GB. A máquina local utilizada é composta por um processador Intel Core i7 de 7º geração, 64 bits e 16 GB de memória RAM rodando no sistema operacional Ubuntu 22.04 LTS.

Para o processamento dos dados, foi necessário definir as *labels*, previamente, num arquivo de extensão ".txt" com a classe do algoritmo - entende-se como classe a letra que o caractere pertence, como "a" ou "b", por exemplo - e as quatro coordenadas que formam a caixa delimitadora para o modelo compreender em que parte da imagem está o caractere que deve ser aprendida. Como no *dataset* original tem apenas as imagens, as *labels* foram adicionadas com a ajuda da aplicação "LabelImg" (Tzutalin, 2015).

O tutorial oficial disponível do YOLOv5 no Google Colab (Ultralytics, 2022) foi utilizado nas etapas de treinamento e validação. O arquivo do modelo YOLOv5 pré-treinado foi utilizado como ponto de partida dos pesos iniciais para o modelo a ser treinado.

3.2. Métricas de Avaliação

Para avaliar este trabalho, utilizou-se a matriz de confusão e as métricas de desempenho precisão (*precision*), *recall* (sensibilidade), *F1-score* e mAP@.5. Precisão é a proporção de Verdadeiro Positivo classificados corretamente; *Recall* é a porcentagem de instâncias positivas que são classificadas como positivas, em outras palavras, o quão bom o classificador é para classificar corretamente a classe de interesse; *F1-score* é a média harmônica ponderada entre a precisão e *recall*, conforme apresentada nas Equações 1, 2, 3:

$$Precisão = \frac{VP}{(VP + FP)} \tag{1}$$

$$Recall = \frac{VP}{(VP + FN)} \tag{2}$$

$$F1 - Score = 2 \cdot \frac{Precisão \cdot Recall}{(Precisão + Recall)} \tag{3}$$

sendo VP, FP e FN os números de Verdadeiros Positivos, Falsos Positivos e Falsos Negativos, respectivamente.

Já o mAP (*mean average precision*) é uma métrica usada em detecção de objetos que representa a média da AP (*average precision*). Para este cálculo, é preciso calcular a Interseção sobre a União (IoU - do inglês *Intersection over Union*), que representa o quão bem as caixas delimitadoras preditas se acoplam nas reais indicadas na fase de rotulagem. Neste trabalho, utilizou-se mAP@.5 (*mean Average Precision at* IoU 0.5), que significa

[‡]https://colab.research.google.com

que apenas os objetos corretamente previstos em caixas delimitadoras com IoU superior a 0,5 são considerados positivos. O mAP@.5 é utilizado como uma medida de quão bem o modelo é capaz de localizar e classificar os objetos corretamente em uma imagem.

4. Resultados e Discussões

4.1. Descrição

As subseções a seguir descrevem os *datasets* e os resultados experimentais do modelo. Os experimentos foram realizados utilizando o conjunto de métricas de desempenho, apresentados na seção anterior, para avaliar a eficácia da abordagem proposta. A matriz de confusão foi utilizada por ser uma técnica que resume o desempenho do algoritmo de classificação, fornecendo uma análise mais detalhada sobre quais classes o modelo está conseguindo acertar ou errar (Brownlee, 2016).

4.2. Descrição do Dataset

Para verificar o impacto no desempenho proveniente do *dataset*, foram realizados vários experimentos usando o *dataset* original, apresentados na Tabela 2. Do *dataset* original, *kaggleDataset*, foram gerados mais dois *datasets*, *smoothRotation* e *noRotation*, formados com caracteres com rotações suaves (de até 45 graus) e sem rotações, respectivamente. Os *datasets* foram divididos em 70% para treino, 20% para validação e 10% para teste. Vale destacar que a divisão do *smoothRotation* é aproximada, devido possuir um quantitativo de caracteres que não permite a divisão exata de 70%, 20% e 10%.

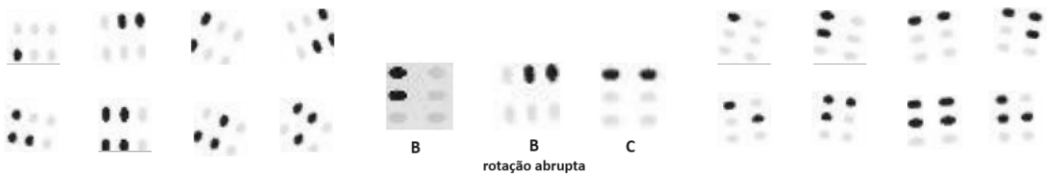
Tabela 2. Dataset

Métodos	Treino	Validação	Teste	Total
kaggleDataset	1.092	312	156	1.560
smoothRotation	806	260	130	1.196
noRotation	728	208	104	1.040

No *kaggleDataset*, todo o *dataset* original foi mantido e apenas separado em treino, teste e validação. Contudo, analisando o *dataset*, foi verificado que existem diversas imagem com rotações abruptas (superiores a 45 graus), como as demostradas na Figura 3(a), com isso, gerando ruídos no treinamento do modelo, ou seja, um caractere pode ser confundido com outro, devido esse acentuado grau de rotação.

Ao analisar o *kaggleDataset*, observou-se que as rotações de um modo geral desempenham um papel influente na detecção e tradução de caracteres Braille. Rotações excessivas podem afetar significativamente a interpretação dos caracteres. Um exemplo disso pode ser percebido na Figura 3(b), quando uma cela Braille representando o caractere “B” possui uma rotação abrupta de 90 graus, se assemelhando ao caractere Braille “C”, gerando ruído no treinamento. Diante disso, retirar esse tipo de rotação tende a melhorar o desempenho do modelo e, para isso, foi gerado mais dois *datasets*, *smoothRotation* e *noRotation*, a partir do *kaggleDataset* para realizar esta análise.

No *dataset* original, cada caractere possui 20 imagens com rotações variadas, de um total de 60 imagens por caractere; no *dataset smoothRotation* são 6 imagens com rotações leves para cada caractere, de um total de 46 imagens por caractere, algumas que foram selecionadas podem ser verificadas na Figura 3(c). Já no *noRotation*, todas as 20 imagens de rotação de cada caractere foram retiradas, deixando somente as variedades de zoom e iluminação. Estes dois *datasets* foram montados para investigar a influência das rotações na detecção e tradução dos caracteres Braille.



(a) Rotações abruptas (b) Ruído gerado por rotação abrupta (c) Rotações suaves

Figura 3. Exemplos de rotações no dataset original

4.3. Resultados

Após o treinamento completo utilizando diferentes configurações de parâmetros aplicados sobre os *datasets*, é apresentado na Figura 4, a Matriz de Confusão para cada *dataset* usado nos experimentos. As linhas representam as classes previstas pelo modelo (*Predicted*) e as colunas representam as classes verdadeiras (*True*), de modo que a predição correta para cada classe se encontra na diagonal principal da matriz, cuja a barra lateral à direita da matriz, indica a variação de valores que oscila de 0 a 100.

Através da matriz de confusão (Figura 4), é possível analisar de maneira mais detalhada os resultados obtidos pelo modelo. A partir da análise experimental, é possível constatar que rotações bruscas (acima de 45 graus) indicam quedas consistentes na precisão de classificação, como é possível observar na Figura 4(a). O impacto de rotações mais condizentes com a realidade, inferiores a 45 graus, pode ser observado na Figura 4(b). Assim, pode-se notar que o *dataset* com rotações suaves melhora significativamente a precisão de reconhecimento, quando comparado ao *dataset* original. Já a Figura 4(c) apresenta o melhor desempenho por não possuir rotações, este caso seria um cenário próximo do ideal, algo difícil de ser praticado no dia-dia.

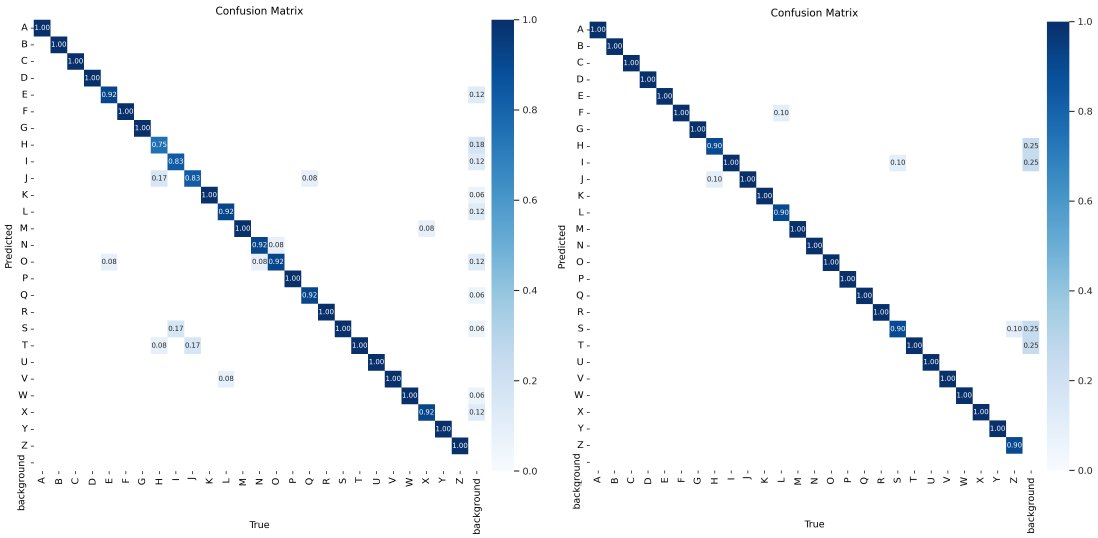
Na Tabela 3, pode-se verificar as métricas provenientes das matrizes de confusão, como Precisão, *Recall*, *F1-score* e mAP@.5. Esses resultados reforçam a capacidade do modelo YOLOv5 em reconhecer células Braille e validam sua aplicabilidade na conversão de caracteres.

Tabela 3. Resultados dos Métodos

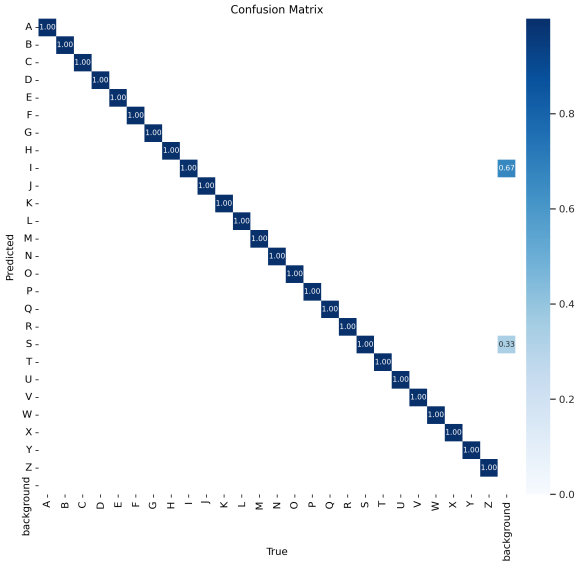
Métodos	Precision	Recall	F1 Score	mAP50
kaggleDataset	0.973	0.969	0.970	0.992
smoothRotation	0.990	0.985	0.990	0.995
noRotation	0.992	0.999	1.000	0.995

Ainda, os resultados (Tabela 3) mostram que a estratégia de utilizar o YOLOv5 supera a abordagem proposta por Kausar em (Kausar *et al.*, 2021). Essa comparação é possível devido a utilização do mesmo *dataset* de imagens, embora tenha usado modelo e métodos diferentes. Enquanto o método com o *dataset* completo deste alcança uma precisão de 88.70%, com 88.20% de *F1-score* e sua técnica com melhor resultado oferece 95.80% de precisão e 95.20% de *F1-score*, os resultados deste trabalho, com o *dataset* completo, mostram uma precisão de 97.30%, *F1-score* de 97.00% e obtendo resultados superiores de 99.00% para precisão e *F1-score* nos *datasets* derivados do original, *smoothRotation* e *noRotation*. Os demais autores apresentados na Seção 2, utilizam *datasets* privados ou distintos deste trabalho, o que dificulta a comparação dos resultados.

O modelo YOLOv5, em sua arquitetura, apresenta a função *Focal Loss* - função



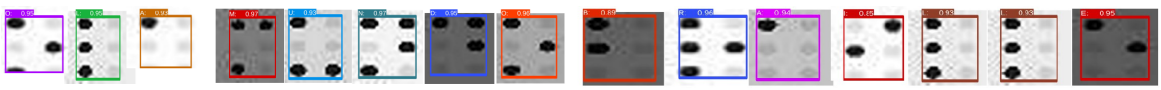
(a) Método KaggleDataset (b) Método smoothRotation



(c) Método noRotation

Figura 4. Matriz de Confusão.

que atribui maior peso aos exemplos difíceis, melhorando a capacidade de detecção e o desempenho geral do modelo - e a estratégia de treinamento em várias escalas. Dessa forma, na Figura 5, é apresentado exemplos de detecções e tradução realizada, sendo que para cada detecção, tem-se a caixa delimitadora, o caractere em português e a taxa de confiança da detecção.



(a) Ola mundo Braille

Figura 5. Reconhecimento e tradução do caractere Braille

De modo geral, o modelo apresentou alto desempenho de reconhecimento Braille, com destaque para o *dataset smoothRotation*, por apresentar rotações mais condizentes

com o que é praticado, por retratar as imagens como são capturadas, normalmente, por um dispositivo móvel. Além de apresentar resultados próximos do *noRotation*, o que demonstra a eficácia do modelo.

5. Considerações Finais

O sistema Braille desempenha um papel crucial para alunos com DV e para aqueles que estão próximos deles. Ao introduzir recursos tecnológicos, como reconhecimento Braille, estes têm o potencial de causar um impacto significativo na comunidade que depende do Braille. Essas inovações permitem que o Braille seja amplamente interpretado e compreendido, abrindo portas para uma comunicação mais inclusiva e facilitando a participação plena desses indivíduos na escola e sociedade.

Dessa forma, a principal contribuição deste trabalho foi desenvolver um método de detecção automática dos caracteres Braille e apresentar a sua correspondente em português com precisão, de modo que possa vir facilitar a comunicação escrita entre professores videntes e alunos com DV. Nessa perspectiva, entende-se que a solução apresentada é expressiva e o uso do detector de objetos YOLOv5, conhecido pela sua alta velocidade de inferência, revelou-se uma boa estratégia para realizar o reconhecimento do Braille.

No entanto, esta abordagem apresenta algumas limitações que podem ser fontes de investigações futuras, como desenvolver uma interface de usuário acessível que possibilite o professor a melhorar a comunicação, tornando mais fácil e rápido fornecer feedback ao aluno com DV, principalmente, no quesito de leitura. Além disso, colegas e responsáveis legais dos estudantes com DV poderão utilizar o sistema. Assim, esse sistema tem potencial de fornecer uma solução aprimorada para a comunicação entre deficientes visuais e videntes.

Agradecimento

Este trabalho foi parcialmente financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) nº 165841/2023-0.

Referências

Albuquerque, D.; Braga, A.; Bomfim, I.; Gomes, D. Aplicando um modelo yolo para detectar e diferenciar por imagem castas de abelhas melíferas de forma automatizada. In: **Anais do XIII Workshop de Computação Aplicada à Gestão do Meio Ambiente e Recursos Naturais**. Porto Alegre, RS, Brasil: SBC, 2022. p. 51–60. ISSN 2595-6124. Disponível em: ⟨https://sol.sbc.org.br/index.php/wcama/article/view/20696⟩.

AlSalman, A.; Gumaei, A.; AlSalman, A.; Al-Hadhrami, S. A Deep Learning-Based Recognition Approach for the Conversion of Multilingual Braille Images. **Computers, Materials Continua**, v. 67, n. 3, p. 3847–3864, 2021. ISSN 1546-2226. Disponível em: ⟨https://www.techscience.com/cmc/v67n3/41628⟩.

Alufaisan, S.; Albur, W.; Alsedrah, S.; Latif, G. Arabic braille numeral recognition using convolutional neural networks. In: Bindhu, V.; Tavares, J. M. R. S.; Boulogeorgos, A.-A. A.; Vuppalapati, C. (Ed.). **International Conference on Communication, Computing and Electronics Systems**. Singapore: Springer Singapore, 2021. p. 87–101. ISBN 978-981-33-4909-4.

Brownlee, J. **What is a Confusion Matrix in Machine Learning**. 2016. Disponível em: ⟨https://machinelearningmastery.com/confusion-matrix-machine-learning/⟩. Acessado em 02 de julho de 2023.

Hsu, B. M. Braille recognition for reducing asymmetric communication between the blind and non-blind. **Symmetry**, v. 12, n. 7, 2020. ISSN 20738994.

Kausar, T. *et al.* Deep Learning Strategy for Braille Character Recognition. **IEEE Access**, IEEE, v. 9, p. 169357–169371, 2021. ISSN 2169-3536. Disponível em: ⟨https://ieeexplore.ieee.org/document/9662337/⟩.

Kubera, E.; Kubik-Komar, A.; Kurasiński, P.; Piotrowska-Weryszko, K.; Skrzypiec, M. Detection and Recognition of Pollen Grains in Multilabel Microscopic Images. **Sensors**, v. 22, n. 7, p. 2690, mar 2022. ISSN 1424-8220. Disponível em: ⟨https://www.mdpi.com/1424-8220/22/7/2690⟩.

Leite, H. **Inclusão na pandemia: como crianças surdas e cegas se adptaram às aulas online**. 2021. Disponível em: ⟨https://paisefilhos.uol.com.br/familia/inclusao-na-pandemia-criancas-surdas-e-cegas-contam-como-se-adaptaram-as-aulas-online/⟩. Acessado em 8 de julho de 2023.

Li, C.; Yan, W. Braille Recognition Using Deep Learning. In: **2021 4th International Conference on Control and Computer Vision**. New York, NY, USA: ACM, 2021. p. 30–35. ISBN 9781450390477. Disponível em: ⟨https://dl.acm.org/doi/10.1145/3484274.3484280⟩.

Ovodov, I. G. Optical braille recognition using object detection neural network. In: **2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)**. [S.l.: s.n.], 2021. p. 1741–1748.

Prakash, S.; Thomas, S.; Gopalan, S. M. Braille Recognition using Convolutional Neural Networks. **Wutan Huatan Jisuan Jishu**, XVI, n. VII, p. 206–213, 2020.

Shanks0465. **Braille Character Dataset**. 2020. Disponível em: ⟨https://www.kaggle.com/datasets/shanks0465/braille-character-dataset⟩. Acessado em 2 de junho de 2023.

Tzutalin. **LabelImg**. 2015. Disponível em: ⟨https://github.com/heartexlabs/labelImg⟩. Acessado em 2 de junho de 2023.

Ultralytics. **YOLOv5 Tutorial**. 2022. Disponível em: ⟨https://colab.research.google.com/github/ultralytics/yolov5/blob/master/tutorial.ipynb⟩. Acessado em 2 de junho de 2023.

Vasanthapriyan, S.; Silva, M. D. Optical Braille Recognition Software Prototype for the Sinhala Language. **Advances in Science, Technology and Engineering Systems Journal**, v. 4, n. 4, p. 221–229, 2019. ISSN 24156698. Disponível em: ⟨https://astesj.com/v04/i04/p27/⟩.